

## PROGRAM

```
//Banker's Algorithm
#include <stdio.h>
struct pro
{
    int all[10], max[10], need[10];
    int flag;
} p[10];

int i, j, pno, r, id, k = 0, safe = 0, exec, count = 0;
int aval[10], seq[10];

void safeState()
{
    while (count != pno)
    {
        for (i = 0; i < pno; i++)
            if (p[i].flag)
            {
                exec = r;
                for (j = 0; j < r; j++)
                    if (p[i].need[j] > aval[j])
                        exec = 0;
                if (exec == r)
                {
                    for (j = 0; j < r; j++)
                        aval[j] += p[i].all[j];
                    p[i].flag = 0;
                    seq[k++] = i;
                    safe = 1;
                    count++;
                }
            }
        if (!safe)
        {
            printf("System is in Unsafe State\n");
            break;
        }
        if (safe)
        {
            printf("System is in Safe State. Safe Sequence: ");
            for (i = 0; i < pno; i++)
                printf("P[%d] \t", seq[i]);
            printf("\n");
        }
    }
}

int main()
```

```
{
    printf("Enter number of process: ");
    scanf("%d", &pno);

    printf("Enter number of resources: ");
    scanf("%d", &r);

    printf("Enter Available resources of each type: ");
    for (j = 0; j < r; j++)
        scanf("%d", &aval[j]);

    printf("Enter Process Details:");
    for (i = 0; i < pno; i++)
    {
        printf("\nProcess %d\n", i);
        printf("Allocation Matrix:\t");
        for (j = 0; j < r; j++)
            scanf("%d", &p[i].all[j]);

        printf("Maximum Matrix:\t\t");
        for (j = 0; j < r; j++)
            scanf("%d", &p[i].max[j]);

        p[i].flag = 1;
        for (j = 0; j < r; j++)
            p[i].need[j] = p[i].max[j] - p[i].all[j];
    }

    printf("\nProcess Details\n");
    printf("PID\t\tAllocation\t\tMax\t\tNeed\n");
    for (i = 0; i < pno; i++)
    {
        printf("%d\t\t", i);
        for (j = 0; j < r; j++)
            printf("%d\t", p[i].all[j]);
        printf("\t\t");
        for (j = 0; j < r; j++)
            printf("%d\t", p[i].max[j]);
        printf("\t\t");
        for (j = 0; j < r; j++)
            printf("%d\t", p[i].need[j]);
        printf("\n");
    }

    safeState();
    return 0;
}
```